MSBD 5002 Homework 3

Pranav A, 20478966

December 1, 2017

1 Classification

1.1 Feature Engineering

It looks like this data is obtained from online MOOCs. An easy way to generate inputs would be to check how much fraction of video has been watched by the student.

Hence I processed the data in such a manner that it would create a vector of 63 dimensions (since 63 videos), which contains the fraction of the video watched.

The total time of video watched could be equivalent to the duration between play and stop or end of browser session. This could be divided by the actual duration of the video which was supplied to us.

However, due to timing constraints, I have only considered if the user has play that particular video or not, and denoted by the 0 or 1.

I used bash scripting to preprocess the data like this.

1.2 Classification Model

The data was divided into 4500 for training and 550 for validation. Then, I used 2-layered neural network for training. To avoid overfitting, I used L2 regularization with cross-entropy loss. Adam was used as the optimizer, with the minibatches of 128 and learning rate of 0.001. The network ran for about 50 epochs, giving an accuracy around 83% on the validation set.

PyTorch with Python 3 was used to create this neural network architecture.

Fractional inputs with better grid search on hyperparameters could have made my model a lot better.

2 Fuzzy clustering with EM algorithm

The formulae given in the slides are only applicable when the points follow Gaussian distribution. Thus it is crucial to convert the points into Gaussian distribution, then apply EM algorithm. This can be done by normalization to zero mean and unit variance. However, in the slides examples, this has been not done like that.

So, I will follow the algorithm according to the slides.

1. For iteration 1, $c_1 = (2.69041963, 5.61486053, 12.61423393, 7.30839493, 0.24305278, 0.99537907)$ and $c_2 = (2.98858857, 7.42781115, 17.13844811, 10.41487571, 0.23788657, 1.00989487)$.
   SSE = 667857.350343.

   For iteration 2, $c_1 = (2.50966266, 4.80465069, 10.40436458, 5.69170016, 0.20847102, 0.94497617)$ and $c_2 = (3.32834732, 8.95352191, 21.51862428, 13.55877955, 0.28751395, 1.09991909)$.
   SSE = 446507.094647

2. Iterations to converge = 29

3. Final updated clusters, $c_1 = (2.40779443, 4.18173286, 8.56845482, 4.47967262, 0.22663779, 0.918)$ and $c_2 = (4.72741511, 15.77470142, 41.31131529, 26.63004261, 0.27119479, 1.42013794)$

With normalization, it only take 13 iterations to converge. Such normalization is necessary, so that one dimension would not dominate over another.

3 Outlier Detection

For $k = 2$, top 5 outliers are, $(526, 67, 679, 403, 334)$ with LOF $(5.416, 4.896, 4.0, 3.727, 3.465)$.
For $k = 3$, top 5 outliers are, $(526, 67, 336, 63, 20)$ with LOF $(4.7780, 4.3154, 2.700, 2.664, 2.5259)$.
The points reported are starting from 1.

Notes

1. The classification code could be similar to the official PyTorch tutorials or from my GitHub which were sloppily and untimely committed.

2. This given solution of the assignment follows the HKUST honour code. Although assignment has been discussed with other peers, the solutions are my own.

3. Kindly give constructive feedback for my incorrect or unclear answers.